

# From homogeneous networks to heterogeneous networks of networks via colored graphlets

John Johnson<sup>1</sup>, Fazle E. Faisal<sup>1,2</sup>, Shawn Gu<sup>1</sup>, and Tijana Milenković<sup>1,2,\*</sup>

<sup>1</sup>Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

<sup>2</sup>Eck Institute for Global Health and Interdisciplinary Center for Network Science and Applications (iCeNSA), University of Notre Dame, Notre Dame, IN 46556, USA

\*To whom correspondence should be addressed (email: tmilenko@nd.edu)

## Abstract

Research of homogeneous (i.e., single node and edge type) biological networks (BNs) has received significant attention. Graphlets have been proven in homogeneous BN research. Given their popularity, graphlets were extended to their directed or dynamic counterparts, owing to increase in availability of directed or temporal BNs. Given the increasing amounts of available BN data of different types, we generalize current homogeneous graphlets to their heterogeneous counterparts, which encompass different node or edge types and thus allow for analyzing a heterogeneous “network of networks”. Heterogeneous graphlets will have at least as high impact as homogeneous graphlets have had.

We illustrate the usefulness of heterogeneous graphlets in the context of network alignment (NA). While existing NA methods are homogeneous (they can only account for a single node type and a single edge type), we generalize a state-of-the-art homogeneous graphlet-based NA method into its heterogeneous counterpart. Because graphlets are applicable to many other network science problems, we provide an implementation of our heterogeneous graphlet counting approach (available upon request).

## 1 Introduction

Owing to advancements of biotechnologies for data collection, various biological network (BN) types exist, such as protein-protein interaction (PPI) or gene co-expression networks. Further, besides genes or proteins, other entities, such as phenotypes (e.g., diseases) or drugs, can be modeled as nodes, and various types of phenotype-phenotype, drug-drug, protein-phenotype, or protein-drug associations can be modeled as edges. Yet, traditional methods for analyzing BNs typically deal with a single homogeneous BN type. BN data integration into a heterogeneous network of networks that encompasses different node or edge types will yield deeper insights into cellular functioning compared to traditional homogeneous BN analyses of individual data types in isolation.

Recent pioneering efforts have recognized the promise of data integration [1]. Yet, these efforts do not account for some of the best practices in homogeneous BN analysis, namely graphlets [2]. Graphlets are small induced subgraphs of a network. They have been proven in homogeneous BN research. They were used as a basis for sensitive measures of network [3] or node [2] similarities. These in turn were used to develop state-of-the-art algorithms for many computational problems, such as network comparison [3] and alignment [4], de-noising (i.e., link prediction) [5], and network clustering [6], as well as for various application problems, such as studying aging [7], cancer and other diseases [8], pathogenicity [6], or receptor-ligand interactions [9]. Importantly, graphlets have been shown to capture topological and functional characteristics of complex real-world networks

better than other approaches, such as network motifs [10], random walks [11], PageRank-like [12] and spectral graph theoretic [13] “topological signatures,” and various centrality measures [7].

Given their popularity, graphlets were extended to their directed [14] or dynamic [15] counterparts, owing to increase in availability of directed or temporal networks. Given the increasing amounts of available heterogeneous BNs, here, we extend current homogeneous graphlets to their heterogeneous counterparts, which encompass multiple node or edge types (i.e., colors; Figure 1).

With the naive exhaustive approach, the number of possible heterogeneous (colored) graphlets increases exponentially with the number of colors [16]. We propose a more efficient colored graphlet approach (Section 2). With our approach, just as with homogeneous graphlets [2], for each node in a heterogeneous network, one can count how many times the given node participates in the given colored graphlet. By doing this for each graphlet (on up to  $n$  nodes), one can form the node’s colored graphlet degree vector (GDV), which quantifies the information about the extended network neighborhood of the node. Given colored GDV for each node in one or more heterogeneous networks, by comparing all GDVs within a network or across networks, one can obtain a measure of similarity between the nodes’ extended network neighborhoods. Then, the resulting node similarities can be used for various network science problems. We illustrate their usefulness in the context of the network alignment problem (Section 3).

## 2 Methods

First, for ease of explanation, we define node-colored graphlets. Given  $k$  possible node colors from set  $C_n = \{c_{n1}, c_{n2}, \dots, c_{nk}\}$ ,  $2^{C_n}$  is the set of all possible combinations of colors from  $C_n$ .  $2^{C_n}$  contains  $\binom{k}{0}$  elements with no color (i.e.,  $\emptyset$ ),  $\binom{k}{1}$  elements with any one color (i.e.,  $\{c_{n1}\}, \{c_{n2}\}, \dots, \{c_{nk}\}$ ),  $\binom{k}{2}$  elements with any two colors (i.e.,  $\{c_{n1}, c_{n2}\}, \{c_{n1}, c_{n3}\}, \dots, \{c_{n1}, c_{nk}\}, \dots, \{c_{nk-1}, c_{nk}\}$ ), and so on. Therefore,  $2^{C_n}$  contains  $\binom{k}{0} + \binom{k}{1} + \binom{k}{2} + \dots + \binom{k}{k} = 2^k$  elements. Now,  $2^{C_n} - \emptyset$  is the set of all possible color combinations from  $C_n$  that excludes the empty set. Therefore,  $2^{C_n} - \emptyset$  contains  $2^k - 1$  elements. Let  $b_n \in 2^{C_n} - \emptyset$ . Given a homogeneous graphlet  $G_i$ , an element  $b_n$  from  $2^{C_n} - \emptyset$ , and the set of colors  $C_n$ , we define a *node-colored graphlet*  $NCG_{i,b_n}$  to be the set of all distinct graphs that are isomorphic to  $G_i$ , while in each such graph, each node is colored with one of the colors from  $b_n$ , and also, each color from  $b_n$  has to be present in each such graph. Thus, given  $k$  node colors, there are  $2^k - 1$  possible node-colored graphlets.

As an illustration, let us assume that a heterogeneous network has nodes with two possible colors:  $c_{n1}$  and  $c_{n2}$ . These two node colors have  $2^2 - 1 = 3$  possible combinations:  $\{c_{n1}\}$ ,  $\{c_{n2}\}$ , and  $\{c_{n1}, c_{n2}\}$ . As a result, for each homogeneous graphlet  $G_i$ , there are three possible node-colored graphlets:  $NCG_{i,\{c_{n1}\}}$ ,  $NCG_{i,\{c_{n2}\}}$ , and  $NCG_{i,\{c_{n1}, c_{n2}\}}$ , where  $NCG_{i,\{c_{n1}\}}$  is a colored version of  $G_i$  that contains only  $c_{n1}$ -colored nodes,  $NCG_{i,\{c_{n2}\}}$  contains only  $c_{n2}$ -colored nodes, and  $NCG_{i,\{c_{n1}, c_{n2}\}}$  contains both  $c_{n1}$ - and  $c_{n2}$ -colored nodes (Fig. 1 (a) illustrates this for a 3-node path, i.e., for homogeneous graphlet  $G_1$ ).

Our definition of node-colored graphlets is more efficient than the naive exhaustive enumerative definition would be. For example, when  $k = 2$  ( $k = 3$ ), with the naive definition, there would be six (18) node-colored graphlets for  $G_1$ , while with our approach there are only three (seven) of them. Even with our approach, the number of node-colored graphlets increases drastically with the increase of  $k$ . Yet, this is not a concern, because in practice, we may expect a relatively small value of  $k$  (e.g., we can study a heterogeneous network whose nodes are proteins, functions, diseases, and drugs with  $k$  value of only four).

Just as an automorphism node orbit of a homogeneous graphlet [2], we define a *node orbit of a node-colored graphlet*  $NCG_{i,b_n}$  as the set of nodes that are “symmetric” to each other in  $NCG_{i,b_n}$

(Fig. 1(a)). Note that our definition of orbits ignores node colors. It is certainly possible to account for colors, but this would increase computational complexity. With our definition, for a homogeneous graphlet with  $x$  orbits, each of its colored graphlets also has  $x$  orbits. There are 73 homogeneous 2-5-node graphlets. So, given  $k$  node colors, there are  $73 \times (2^k - 1)$  orbits for 2-5-node node-colored graphlets. We define *node-colored GDV* as a vector containing counts of how many times the given node “touches” each node-colored graphlet at each orbit.

Second, analogous to node-colored graphlets, without going again through all the formalisms due to space constraints, we define *edge-colored graphlets* (Fig. 1 (b)), *node orbits in edge-colored graphlets*, and *edge-colored GDV*. Given  $j$  edge colors, there are  $2^j - 1$  possible edge-colored graphlets. Yet, we can study e.g., a network whose nodes are genes/proteins and whose edges are PPIs, genetic interactions, gene co-expressions, and signaling interactions with only four edge colors.

Third, the above ideas can be combined into graphlets that have different node as well as edge colors. A computationally simple option is to concatenate node- and edge-colored GDVs. A more complex option is, for each node-colored graphlet, to vary its edge colors (or vice versa). While this would further increase the number of colored graphlets of interest, innovative approaches have been introduced for faster homogeneous graphlet counting, even in networks with millions of nodes counting [17, 18], and similar directions can be pursued for heterogeneous graphlet counting.

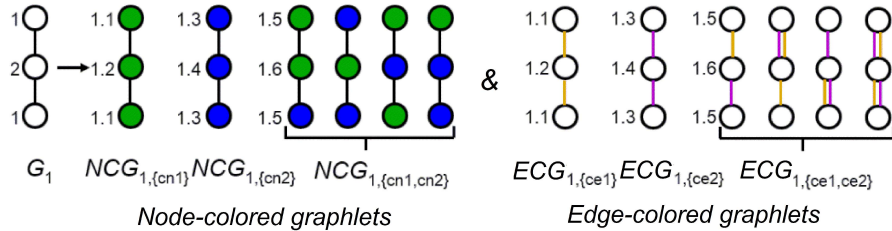


Figure 1: Illustration of node-colored (left) and edge-colored (right) graphlets. Given two node colors  $c_{n1}$  (green) and  $c_{n2}$  (blue), the homogeneous graphlet  $G_1$  results in three node-colored graphlets:  $NCG_{1,\{c_{n1}\}}$ ,  $NCG_{1,\{c_{n2}\}}$ , and  $NCG_{1,\{c_{n1},c_{n2}\}}$ .  $G_1$  has two node orbits:  $\{1,2\}$ . Similarly, each of  $NCG_{1,\{c_{n1}\}}$ ,  $NCG_{1,\{c_{n2}\}}$ , and  $NCG_{1,\{c_{n1},c_{n2}\}}$  has two orbits, independent of node colors:  $\{1.1, 1.2\}$ ,  $\{1.3, 1.4\}$ , and  $\{1.5, 1.6\}$ , respectively. Analogous to the above, given two edge colors  $c_{e1}$  (orange) and  $c_{e2}$  (purple),  $G_1$  results in three edge-colored graphlets ( $ECG_{1,\{c_{e1}\}}$ ,  $ECG_{1,\{c_{e2}\}}$ , and  $ECG_{1,\{c_{e1},c_{e2}\}}$ ), each with two node orbits.

### 3 Results and discussion

We illustrate an application of (node-)colored graphlets in the context of a popular network alignment (NA) problem. NA aims to find a mapping between nodes of compared networks that identifies regions of similarities between the networks. NA can align biological networks of different species, to allow for transferring biological knowledge from well-studied to poorly-studied species between the aligned network regions [19]. As a proof of concept, we modify an existing homogeneous state-of-the-art NA method called WAVE [4], which happens to be graphlet-based [19], to account for node-colored graphlets (as described in Supplementary Section S1). We hypothesize that when our new heterogeneous WAVE and its existing homogeneous version are both run on the same heterogeneous node-colored network data (where every aspect of the analysis remains the same except that the former accounts for the different node colors while the latter ignores them), heterogeneous NA

will outperform homogeneous NA. Note that existing NA methods can only deal with homogeneous network data. So, we advance the NA field with our new heterogeneous NA approach.

We denote by  $\text{WAVE}_{1C}$  the existing homogeneous WAVE that deals with one node color, and by  $\text{WAVE}_{2C}$ ,  $\text{WAVE}_{3C}$ , and  $\text{WAVE}_{4C}$  our heterogeneous WAVE that deals with two, three, and four node colors.

We evaluate homogeneous versus heterogeneous WAVE versions in three tests. First, we align synthetic networks, which originate from three different network models (GEO, SF, and ER) and which have up to four artificially imposed node colors (Supplementary Section S2.1), to their noisy counterparts (defined below). Second, we align homogeneous (single-node-type) PPI networks, which have up to four node colors imposed according to proteins’ involvement in a combination of aging, cancer, and Alzheimer disease (Supplementary Section S2.2.1), to their noisy counterparts (defined below). Here, node colors originate from gene expression (Expr) or sequence (Seq) analyses. Also, here, we consider each of three types of PPIs: only affinity capture coupled to mass spectrometry (APMS), only two-hybrid (Y2H), and both combined (APMS+Y2H). Third, we align heterogeneous (two-node-type) biological networks (Supplementary Section S2.2.2) to their noisy counterparts (defined below). Here, the two node types/colors correspond to proteins and their Gene Ontology (GO) terms, and edges exist between proteins (where we again use each of the three types of PPIs), between GO terms, and between proteins and GO terms (Supplementary Section S2.2.2).

In all three tests, a noisy counterpart is a copy of the original network with  $x\%$  of the edges randomly rewired, where we vary the noise level  $x$  from 10% to 50% in increments of 10% (Supplementary Section S2.3). Because noisy versions are just the original network rewired, we know the true node mapping between the aligned networks (this mapping is hidden from each NA method when it is asked to produce an alignment). Therefore, we evaluate the quality of the given alignment by measuring its node correctness (NC), which quantifies how well the alignment matches the true node mapping. Formally, NC is the percentage of node pairs from the given alignment that are present in the true node mapping.

We use the above evaluation framework to test whether heterogeneous NA improves upon the traditional homogeneous NA, i.e., whether increasing the number of node colors within WAVE results in improved alignment quality. Indeed, this is exactly what we observe in all three tests and across all noise levels (Fig. 2).

## 4 Conclusion

We show the power of heterogeneous graphlets in the context of the network alignment problem. Because graphlets are useful for other network science problems, we provide an implementation of our heterogeneous graphlet counting approach (available upon request).

## Funding

This work was supported by the National Science Foundation [CCF-1319469, CAREER CCF-1452795].

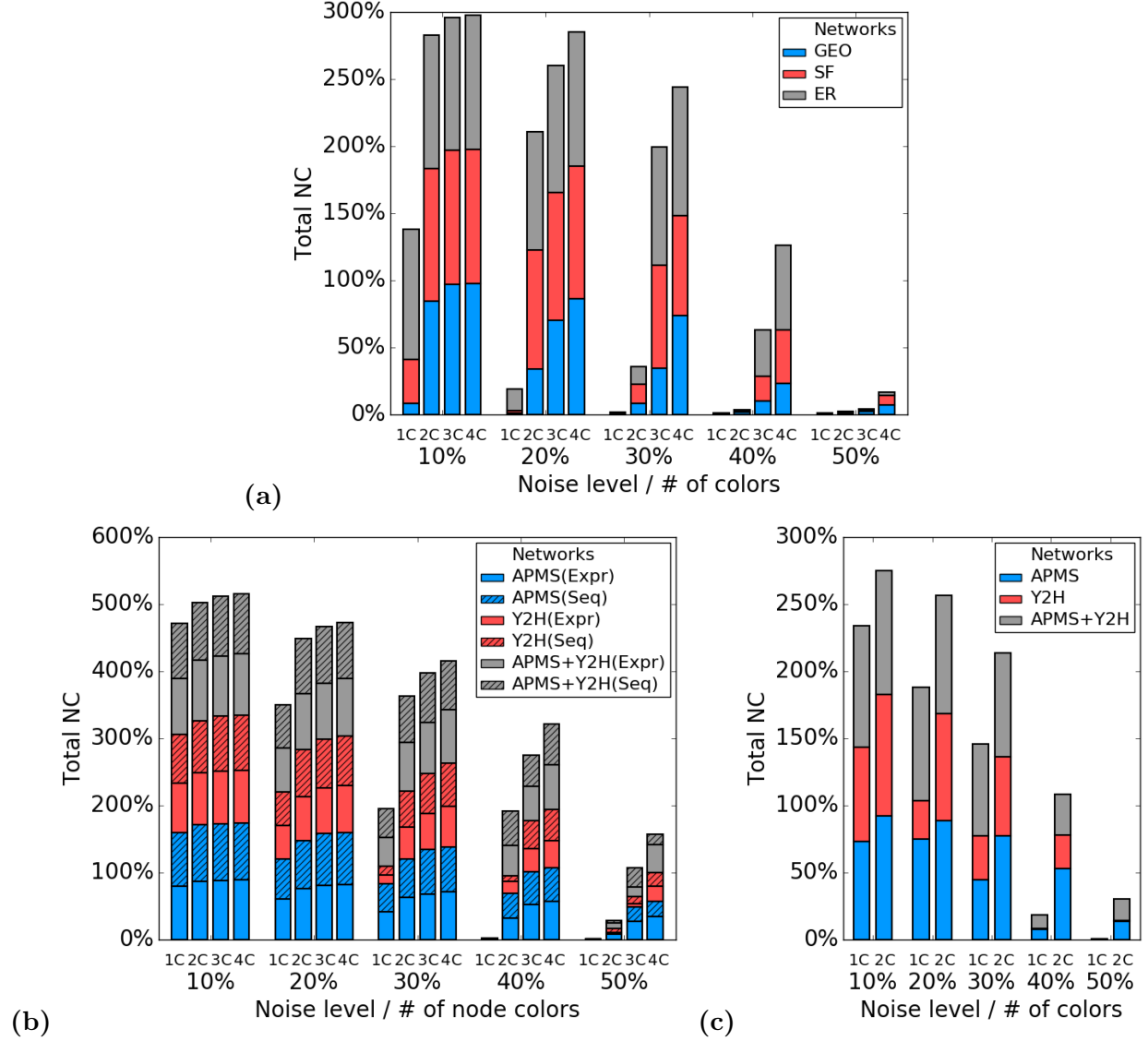


Figure 2: Node correctness (totaled over all analyzed networks in the given test) vs. Noise level / Number of colors, in **(a)** evaluation test one (synthetic networks), **(b)** evaluation test two (PPI networks), and **(c)** evaluation test three (Protein-GO networks). See the text for details. Note that “1C” corresponds to homogeneous  $WAVE_{1C}$ , “2C” corresponds to heterogeneous  $WAVE_{2C}$  with two colors, “3C” corresponds to heterogeneous  $WAVE_{3C}$  with three colors, and “4C” corresponds to heterogeneous  $WAVE_{4C}$  with four colors.

# SUPPLEMENTARY INFORMATION FOR: From homogeneous networks to heterogeneous networks of networks via colored graphlets

John Johnson<sup>1</sup>, Fazle E. Faisal<sup>1,2</sup>, Shawn Gu<sup>1</sup>, and Tijana Milenković<sup>1,2,\*</sup>

<sup>1</sup>Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

<sup>2</sup>Eck Institute for Global Health and Interdisciplinary Center for Network Science and Applications (iCeNSA), University of Notre Dame, Notre Dame, IN 46556, USA

\*To whom correspondence should be addressed (email: tmilenko@nd.edu)

## S1 From homogeneous WAVE<sub>1C</sub> to heterogeneous WAVE<sub>2C+</sub>

Homogeneous WAVE<sub>1C</sub> [4] has two parts.

First, a node cost function (NCF) captures pairwise similarities between nodes of the two networks being aligned. In particular, WAVE<sub>1C</sub>'s NCF computes the topological similarity between extended network neighborhoods of two nodes using homogeneous graphlets. Specifically, it first summarizes the extended network neighborhood of a node into its graphlet degree vector (GDV), which counts how many times the given node “touches” each of homogeneous graphlets on up to five nodes, i.e., each of their 73 orbits. Then, it compares two nodes' GDVs to quantify the topological similarity between the two nodes. The GDV similarity is computed as follows. We denote the set of all nodes' (homogeneous) GDVs (for both networks) as the GDV matrix. To compute pairwise node similarities across the two networks, we first perform principal component analysis (PCA) on this matrix. Then, we pick the first  $m$  principal components, where the value of  $m$  is at least two and as low as possible so that the  $m$  components account for at least 90% variation in the data. Finally, for each pair of nodes from the networks being aligned, we compute their cosine similarity based on the nodes' first  $m$  principal components. The resulting pairwise node similarities are WAVE<sub>1C</sub>'s homogeneous NCF.

Second, this NCF is fed into WAVE<sub>1C</sub>'s alignment strategy (AS) to construct an alignment. The AS aims to optimize both the total NCF similarity over all aligned nodes (also known as node conservation) as well as edge conservation while constructing an alignment. Starting with a pair of highly NCF-similar nodes, called seed nodes, the AS iteratively seeds-and-expands around the seed in a greedy fashion. Given the current alignment (originally the seed), the AS tries to align neighbors of each node in the current alignment (thus optimizing edge conservation) while at the same time favoring conserving edges whose end nodes are NCF-similar (thus optimizing node conservation). This continues until an entire alignment is constructed.

Analogous to WAVE<sub>1C</sub>, WAVE<sub>2C+</sub> (i.e., WAVE<sub>2C</sub>, WAVE<sub>3C</sub>, or WAVE<sub>4C</sub>) computes *node-colored* GDV for each node in each of the two heterogeneous networks being aligned. We denote the set of all nodes' *colored* GDVs (for both networks) as the colored GDV matrix. To compute pairwise node similarities across the two networks, we first perform principal component analysis (PCA) on this matrix. Then, we pick the first  $m$  principal components, where the value of  $m$  is at least two and as low as possible so that the  $m$  components account for at least 90% variation in the data. Finally, for each pair of nodes from the networks being aligned, we compute their cosine similarity based on the nodes' first  $m$  principal components. The resulting pairwise node similarities are our new heterogeneous NCF.

Then, WAVE<sub>2C+</sub> (i.e., WAVE<sub>2C</sub>, WAVE<sub>3C</sub>, or WAVE<sub>4C</sub>) uses the AS of WAVE<sub>1C</sub>. This is because we want to favor aligning two nodes of the same color over aligning (equally NCF-similar) nodes of different colors. For example, when aligning two protein-GO networks, we want to map

proteins to proteins and GO terms to GO terms, and we do not want to map proteins to GO terms. This similarity is captured in the heterogeneous NCF, allowing us to use the same AS that WAVE<sub>1C</sub> uses, which results in the heterogeneous WAVE<sub>2C+</sub> versions.

## S2 Forming networks

We compare the heterogeneous WAVE<sub>2C+</sub> versions (i.e., WAVE<sub>2C</sub>, WAVE<sub>3C</sub>, and WAVE<sub>4C</sub>) to the homogeneous WAVE<sub>1C</sub> on: 1) synthetic networks with up to four artificially imposed node colors (Supplementary Section S2.1), 2) homogeneous (single-node-type) PPI networks that have up to four node colors imposed according to proteins’ involvement in a combination of aging, cancer, and Alzheimer disease (AD) (Supplementary Section S2.2.1), and 3) heterogeneous (two-node-type) biological networks, where the two node colors correspond to proteins and their Gene Ontology (GO) terms, and edges exist between proteins, between GO terms, and between proteins and GO terms (Supplementary Section S2.2.2). We align each of the above networks to its noisy version, as described in Supplementary Section S2.3. Next, we explain how we form the three groups of networks.

### S2.1 Synthetic networks

We form synthetic networks using three random graph generators, namely: 1) geometric random graphs (GEO), 2) scale-free networks (SF), and 3) Erdős-Rényi random graphs (ER), and we form five random network instances per model and average results over the five instances. The three models have distinct network topologies [20], which enables us to test the robustness of our results to the choice of random graph model. We set all three model network instances to the same size (1,000 nodes and 6,000 edges). Since the existing random graph generators are not designed to produce heterogeneous networks, we simply randomly assign each node a color out of  $k$  possible colors, where there are approximately  $1,000/k$  nodes of each color. We vary  $k$  from one to four. That is, for each synthetic network instance, we form four of its heterogeneous versions, each with one of the following number of colors: 1) one color, 2) two colors, 3) three colors, and 4) four colors.

### S2.2 Real-world networks

#### S2.2.1 Homogeneous PPI networks with node colors imposed according to proteins’ involvement in certain biological processes

We obtain the human PPI network data from BioGRID [21]. We consider three different human PPI networks, one with each of the following types of physical interactions: affinity capture coupled to mass spectrometry (APMS) PPIs only, yeast two-hybrid (Y2H) PPIs only, and APMS and Y2H PPIs combined. We keep only the largest connected components of the three networks. The resulting APMS network has 11,450 nodes and 92,257 edges. The resulting Y2H network has 10,317 nodes and 41,925 edges. The resulting APMS+Y2H network has 14,314 nodes and 132,657 edges.

We impose node colors onto each PPI network according to the proteins’ involvement in certain biological processes, namely a combination of aging, cancer, or Alzheimer disease (AD), as explained below. The data originate from the following sources. We obtain a list of sequence-based (Seq) human aging-related genes from GenAge [22] and a list of gene expression-based (Exp) human aging-related genes from the study by [23]. We obtain a list of genes implicated in cancer from COSMIC [24]. We obtain a list of human genes related to AD from [25].

We use these data to impose colors onto nodes (i.e., proteins) in each of the three PPI networks as follows. For the given network, we use sequence-based (Seq) aging- and cancer-related data to form four different colored versions of the network (analogous to colored versions of a synthetic network), as described below.

- In the 1-colored network, we treat all the nodes the same, meaning that they all have the same color.
- In the 2-colored network, we use the aging-related data to color nodes as “aging-related”. Otherwise, if a node is not in the aging-related data, we color it as “non-aging-related”. In this way, we color 270 nodes as “aging-related” and 10,047 nodes as “non-aging-related”.
- In the 3-colored network, we use the aging- and cancer-related data to color nodes as follows. If a node is present in the aging-related data, we color it as “aging-related”. If a node is absent from the aging-related data but is present in the cancer-related data, we color it as “cancer only”. If a node is absent from both of the data sets, we color it as “non-aging-related and non-cancer”. In this way, we color 270 nodes as “aging-related”, 405 nodes as “cancer only”, and 9,642 nodes as “non-aging-related and non-cancer”.
- In the 4-colored network, we use the aging- and cancer-related data to color nodes as follows. If a node is absent from the cancer-related data but is present in the aging-related data, we color it as “aging-related only”. If a node is absent from the aging-related data but is present in the cancer-related data, we color it as “cancer only”. If a node is present in both of the data sets, we color it as “both aging-related and cancer”. If a node is absent from both of the data sets, we color it as “non-aging-related and non-cancer”. In this way, we color 203 nodes as “aging-related only”, 405 nodes as “cancer only”, 67 nodes as “both aging-related and cancer”, and 9,642 nodes as “non-aging-related and non-cancer”.

To test the robustness of the choice of node color data used above, we vary the underlying data, as follows. Now, for each of the three PPI networks, we use expression-based (Exp) aging- and AD-related data to form four different colored versions of the given network, as described below.

- In the 1-colored network, we treat all the nodes the same, meaning that they all have the same color.
- In the 2-colored network, we use the aging-related data to color nodes as “aging-related”. Otherwise, if a node is not in the aging-related data, we color it as “non-aging-related”. In this way, we color 2,889 nodes as “aging-related” and 7,428 nodes as “non-aging-related”.
- In the 3-colored network, we use the aging- and AD-related data to color nodes as follows. If a node is present in the aging-related data, we color it as “aging-related”. If a node is absent from the aging-related data but is present in the AD-related data, we color it as “AD only”. If a node is absent from both of the data sets, we color it as “non-aging-related and non-AD”. In this way, we color 2,889 nodes as “aging-related”, 356 nodes as “AD only”, and 7,072 nodes as “non-aging-related and non-AD”.
- In the 4-colored network, we use the aging- and AD-related data to color nodes as follows. If a node is absent from the AD-related data but is present in the aging-related data, we color it as “aging-related only”. If a node is absent from the aging-related data but is present in the AD-related data, we color it as “AD only”. If a node is present in both of the data



sets, we color it as “both aging-related and AD”. If a node is absent from both of the data sets, we color it as “non-aging-related and non-AD”. In this way, we color 2,232 nodes as “aging-related only”, 356 nodes as “AD only”, 657 nodes as “both aging-related and AD”, and 7,072 nodes as “non-aging-related and non-AD”.

### S2.2.2 Heterogeneous real-world protein-GO networks

A heterogeneous protein-GO network has two types of nodes: protein and GO term (henceforth, we simply refer to the latter as GO) and three types of edges: 1) PPI, 2) protein-GO association, and 3) GO-GO semantic similarity. The PPI data are the same three types of PPI networks as in Supplementary Section S2.2.1 (i.e., APMS, Y2H, APMS+Y2H), protein-GO associations are obtained from the Gene Ontology Consortium [26] based on experimental evidence codes, and GO-GO semantic similarities are computed as follows. We compute semantic similarity between all GOs that annotate at least one protein in the given considered PPI network. We use Lin method [27] to compute the semantic similarity. We form edges between GOs using semantic similarity threshold of 0.7, because the density of the resulting GO-GO network approximately matches the density of the PPI network.

Considering APMS PPIs only, Y2H PPIs only, and both APMS and Y2H PPIs, we form three heterogeneous protein-GO networks for human, whose sizes are shown in Supplementary Tables S1 and S2.

Network type	Node type		
	Protein	GO	All combined
APMS	11,450	5,558	17,008
Y2H	10,317	5,554	15,871
APMS+Y2H	14,314	6,126	20,440

Table S1: The number of nodes in the three heterogeneous protein-GO networks.

Network type	Edge type			
	PPI	Protein-GO	GO-GO	All combined
APMS	92,257	24,854	48,731	165,842
Y2H	41,925	24,473	48,873	115,271
APMS+Y2H	132,657	29,476	56,313	218,446

Table S2: The number of edges in the three heterogeneous protein-GO networks.

## S2.3 Creating noisy counterparts of synthetic and real-world networks

Given an original synthetic or real-world network  $G = (V, E)$ , we construct a noisy counterpart of the original network as follows. Considering a noise level of  $x\%$ , we randomly choose  $x\%$  of the edges and remove them from the original network, and then we randomly choose the same number of node pairs that are not connected in the original network and add edges between these nodes. That is, we randomly rewired  $x\%$  of the edges in the original network. Clearly, each randomly rewired noisy network will have the same number of nodes and edges as the original network. For each considered original network, we consider the following noise levels: 10%, 20%, 30%, 40%, and

50%. We construct five noisy networks at each noise level to account for the randomness in edge rewiring; then, for each noise level, we report results averaged over the five random runs.

## References

- [1] V. Gligorijević and N. Pržulj. Methods for biological data integration: perspectives and challenges. *Journal of The Royal Society Interface*, 12(112):20150571, 2015.
- [2] T. Milenković and N. Pržulj. Uncovering biological network function via graphlet degree signatures. *Cancer Informatics*, 6:257–273, 2008.
- [3] Ö.N. Yaveroğlu, T. Milenković, and N. Pržulj. Proper evaluation of alignment-free network comparison methods. *Bioinformatics*, 31(16):2697–2704, 2015.
- [4] Y. Sun, J. Crawford, J. Tang, and T. Milenković. Simultaneous optimization of both node and edge conservation in network alignment via WAVE. In *International Workshop on Algorithms in Bioinformatics*, pages 16–39. Springer, 2015.
- [5] Y. Hulovatyy, R.W. Solava, and T. Milenković. Revealing missing parts of the interactome via link prediction. *PLOS ONE*, 9(3):e90073, 2014.
- [6] R.W. Solava, R.P. Michaels, and T. Milenković. Graphlet-based edge clustering reveals pathogen-interacting proteins. *Bioinformatics*, 18(28):i480–i486, 2012.
- [7] F.E. Faisal and T. Milenković. Dynamic networks reveal key players in aging. *Bioinformatics*, 30:1721–1729, 2014.
- [8] X.D. Wang, J.L. Huang, L. Yang, D.Q. Wei, Y.X. Qi, and Z.L. Jiang. Identification of human disease genes from interactome network using graphlet interaction. *PLOS ONE*, 9(1):e86142, 2014.
- [9] O. Singh, K. Sawariya, and P. Aparoy. Graphlet signature-based scoring method to estimate protein–ligand binding affinity. *Royal Society Open Science*, 1(4):140306, 2014.
- [10] N. Pržulj. Protein-protein interactions: Making sense of networks via graph-theoretic modeling. *Bioessays*, 33(2):115–123, 2011.
- [11] T. Milenković, V. Memišević, A. Bonato, and N. Pržulj. Dominating biological networks. *PLOS ONE*, 6(8):e23016, 2011.
- [12] F.E. Faisal, H. Zhao, and T. Milenković. Global network alignment in the context of aging. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 12(1):40–52, 2014.
- [13] J. Crawford, Y. Sun, and T. Milenković. Fair evaluation of global network aligners. *Algorithms for Molecular Biology*, 10(1):1, 2015.
- [14] A. Sarajlić, N. Malod-Dognin, Ö.N. Yaveroğlu, and N. Pržulj. Graphlet-based characterization of directed networks. *Scientific Reports*, 6, 2016.
- [15] Y. Hulovatyy, H. Chen, and T. Milenković. Exploring the structure and function of temporal networks with dynamic graphlets. *Bioinformatics*, 31(12):i171–i180, 2015.

- [16] V. Vacic, L.M. Iakoucheva, S. Lonardi, and P. Radivojac. Graphlet kernels for prediction of functional residues in protein structures. *Journal of Computational Biology*, 17(1):55–72, 2010.
- [17] N.K. Ahmed, J. Neville, R.A. Rossi, and N. Duffield. Efficient graphlet counting for large networks. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 1–10. IEEE, 2015.
- [18] R.A. Rossi and R. Zhou. Hybrid CPU-GPU Framework for Network Motifs. *arXiv preprint arXiv:1608.05138*, 2016.
- [19] L. Meng, A. Striegel, and T. Milenković. Local versus global biological network alignment. *Bioinformatics*, 32(20):3155–3164, 2016.
- [20] T. Milenković, J. Lai, and N. Pržulj. GraphCrunch: a tool for large network analyses. *BMC Bioinformatics*, 9(70), 2008.
- [21] B. J. Breitkreutz, C. Stark, , T. Reguly, L. Boucher, A. Breitkreutz, M. Livstone, R. Oughtred, D. H. Lackner, J. Bahler, V. Wood, K. Dolinski, and M. Tyers. The BioGRID Interaction Database: 2008 update. *Nucleic Acids Research*, 36:D637–D640, 2008.
- [22] J.P. de Magalhães. Aging research in the post-genome era: New technologies for an old problem. In C.H. Foyer, R. Faragher, and P.J. Thornalley, editors, *Redox Metabolism and Longevity Relationships in Animals and Plants*, pages 99–115. Taylor and Francis, New York, 2009.
- [23] N.C. Berchtold, D.H. Cribbs, P.D. Coleman, J. Rogers, E. Head, R. Kim, T. Beach, C. Miller, J. Troncoso, J.Q. Trojanowski, H.R. Zielke, and C.W. Cotman. Gene expression changes in the course of normal brain aging are sexually dimorphic. *Proceedings of the National Academy of Sciences*, 105(40):15605–10, 2008.
- [24] Malachi Griffith and Obi L. Griffith. Cosmic (catalogue of somatic mutations in cancer). *Dictionary of Bioinformatics and Computational Biology*, 2004.
- [25] Julie Simpson, Paul Ince, Paul Heath, Rohini Raman, Claire Garwood, Pamela Shaw, Catherine Gelstorpe, Lynne Baxter, Gillian Forster, Matthews Fiona, and et al. Microarray analysis of the astrocyte transcriptome in the aging brain: Relationship to alzheimer’s pathology and apoe genotype. *Alzheimer’s & Dementia*, 7(4), 2011.
- [26] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, and et al. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):2529, 2000.
- [27] Gaston K. Mazandu and Nicola J. Mulder. Dago-fun: tool for gene ontology-based functional analysis using term information content measures. *BMC Bioinformatics*, 14(1):284, 2013.